

УДК 338.001.36

О. И. Еськова (oieskova@rambler.ru),
канд. техн. наук, доцент

Д. Н. Каптур (Scpobject173@gmail.com),
студент

Белорусский торгово-экономический
университет потребительской кооперации
г. Гомель, Республика Беларусь

ТЕХНОЛОГИИ РАЗРАБОТКИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ИНТЕРНЕТ-МАГАЗИНА ДЛЯ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ

В данной статье выполнен обзор технологий, которые были использованы при создании мобильного приложения интернет-магазина для потребительской кооперации.

This article provides an overview of the technologies that were used to create a mobile application for an online store for consumer cooperation.

Ключевые слова: мобильное приложение; интернет-магазин; технология разработки; Java; плагин; фреймворк; база данных; формат JSON; Web-сервер; мобильный клиент; PHP; JavaScript; шаблоны; интерфейс; администрирование; API.

Key words: mobile application; online store; development technology; Java; plugin; framework; database; format JSON; Web-server; mobile client; PHP; JavaScript; templates; interface; administration; API.

Разработка мобильных приложений по своему определению производится в условиях различных ограничений: на объем памяти мобильного устройства, на степень новизны операционной системы, на наличие или отсутствие устойчивого интернет-соединения, на скорость реакции устройства при взаимодействии с пользователем и т. д. Но особенно серьезными эти ограничения становятся при разработке интернет-магазина, целевой аудиторией которого являются люди, проживающие в сельской местности. Основным способом доступа к Сети для них являются мобильные сети стандарта 2G и 3G, покрытие которых даже сегодня не является полным на всей территории Республики Беларусь. При этом сами мобильные устройства представлены бюджетными аппаратами с малым объемом памяти. Поэтому очень важен выбор технологий разработки приложения и реализации клиент-серверного взаимодействия, которые наиболее простым и эффективным способом обеспечат учет всех ограничений.

Выбор способа реализации серверной части и хранения базы данных магазина оказывает влияние не только на производительность приложения и надежность получаемых клиентами данных, но и на возможность развития и масштабирования данного приложения.

Лидером в решении данной задачи являются облачные платформы. Примером такой платформы является Firebase [1]. Данные хранятся в формате JSON и синхронизируются в реальном времени с каждым подключенным клиентом. Приложения Firebase остаются отзывчивыми даже в автономном режиме, поскольку данные клиентов сохраняются на диске. После восстановления подключения клиентское устройство получает все пропущенные изменения, синхронизируя их с текущим состоянием сервера.

Однако использование Firebase имеет свои недостатки. По сравнению с классическим языком SQL возможность создания запросов и множественной записи в Firebase весьма ограничена. Далеко не со всеми структурами данных удобно работать. Существует ограничение до 100 подключений и 1 Гб хранилища при использовании этого инструмента. Хотя разработчики Firebase декларируют, что ее использование не требует создания отдельного сервера приложения, на деле в большинстве случаев это оказывается далеко не так, и требуется использование дополнительных сервисов, таких как Algolia или Elasticsearch. Поэтому создание собственного веб-сервера со встроенным API может быть более гибким и продуманным решением, особенно в том случае, когда требуется специфическая обработка или оптимизация данных на стороне сервера.

В итоге для серверной части интернет-магазина был выбран подход с созданием собственного серверного приложения, которое использует классическую реляционную базу данных MySQL. Возможность создания запросов в этом случае имеет достаточно богатый спектр. Но после формирования результата и перед отправкой клиенту он преобразуется в формат JSON. Поэтому клиентская часть приложения не зависит от способа реализации серверной части, что позволит при необходимости в дальнейшем использовать сторонние сервисы для хранения ба-

зы данных магазина. Такой подход выигрывает в адаптивности к специфическим условиям работы приложения, но в тоже время проигрывает в сложности его создания.

Web-сервер интернет-магазина в основном был реализован на языке PHP. Этот язык специально сконструирован для веб-разработок, и его код может внедряться непосредственно в HTML [2]. PHP-скрипты выполняются на сервере и генерируют HTML, который посылается клиенту. Для интерфейса панели администратора использовался Java Script, поскольку он дает возможность использовать ряд мощных современных шаблонов для администрирования (таких как Isomorphic, Clean UI React, Jumbo React Admin и др.). Использование готовых шаблонов значительно упрощает программирование, обеспечивая удобный и эффектный интерфейс (рисунок 1).

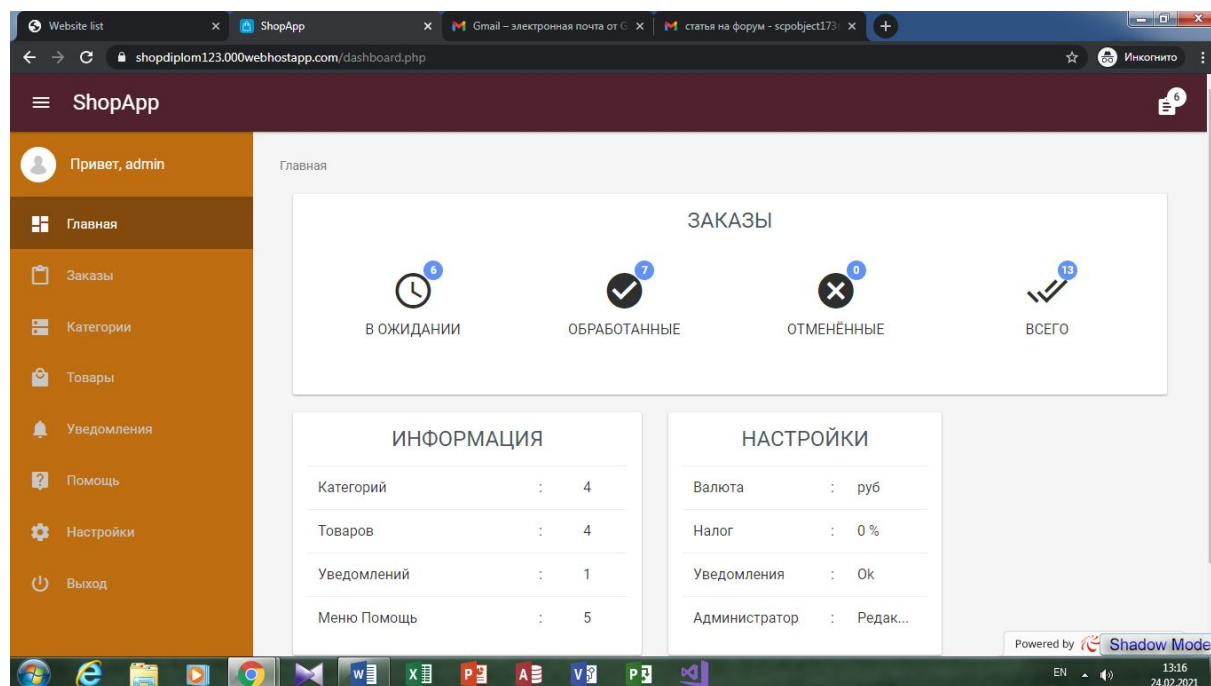


Рисунок 1 – Интерфейс панели администратора магазина

Для разработки Web-сервера использовались также библиотеки и плагины bootstrap, jQuery и TinyMCE. Bootstrap – это популярный бесплатный набор инструментов с открытым исходным кодом для создания и настройки адаптивных сайтов. Благодаря этому фреймворку было организовано комфортное для пользователя взаимодействие через формы ввода данных.

Библиотека jQuery – это простое дополнение к JavaScript, позволяющее создавать различные визуальные эффекты (растворяющиеся элементы, связанная анимация, выдвижные панели и т. п.) [3]. В программе использовалась для создания анимации при взаимодействии с меню разделов панели администратора, перехода между разделами, выпадающих меню.

Редактор TinyMCE служит для создания и наполнения содержимым Web-сайтов в режиме WYSIWYG («Что вижу – то и получаю»). Редактор позволяет вставлять рисунки, таблицы, указывать стили оформления текста, видео. Используется в различных CMS-системах и поддерживает различные темы, шаблоны и языки, а также имеет возможность подключения модулей (плагинов). Редактор TinyMCE также используется для создания красивых и привлекательных описаний товаров с возможностью вставки видео- и аудиофайлов, таблиц, изображений.

Мобильный клиент интернет-магазина написан на языке Java с использованием библиотеки Onesignal для создания уведомлений, библиотек GSON и Volley для конвертации формата JSON в объекты Java и обратно. Кроме того, для создания изображений с круглыми углами использовалась библиотека RoundedImageView, а для создания эффекта визуальной ряби при прикосновении пользователя был применен компонент Ripple современного дизайна Material [4]. Пример интерфейса пользователя для интернет-магазина показан на рисунке 2.

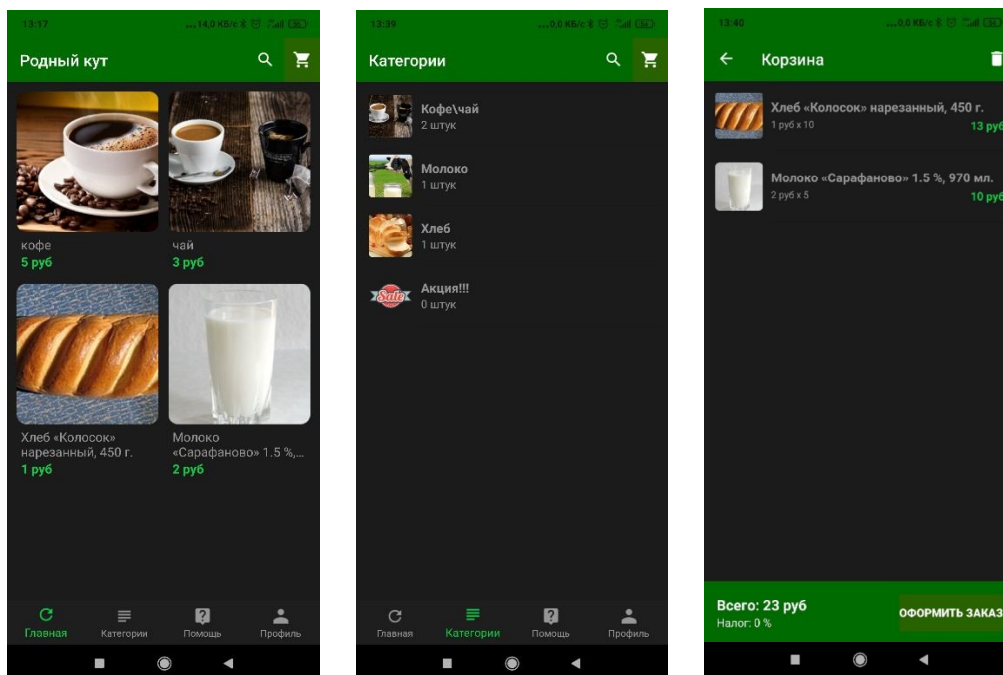


Рисунок 2 – Интерфейс мобильного клиента магазина

Таким образом, разработка мобильной версии интернет-магазина требует использования широкого спектра технологий разработки, каждая из которых выполняет свою специфическую задачу. Эти технологии должны быть совместимы, эффективно дополнять друг друга и при этом обеспечивать возможность развития и масштабирования приложения.

Список использованной литературы

1. **Firestore** помогает создавать и запускать успешные приложения [Электронный ресурс]. – Режим доступа : <https://firebase.google.com>. – Дата доступа : 27.02.2021.
2. **Руководство** по PHP [Электронный ресурс]. – Режим доступа : <https://www.php.net/manual/ru/index.php>. – Дата доступа : 27.02.2021.
3. **Установка** и настройка TinyMCE [Электронный ресурс]. – Режим доступа : <https://r-band.ru/poleznosti/ustanovka-i-nastroyka-tinymce>. – Дата доступа : 27.02.2021.
4. **Tutorials** Material Design [Electronic resource]. – Mode of access : <https://material.io/resources/tutorials>. – Date of access : 27.02.2021.